HPC Profiler

Nsight Systems and Nsight Compute

OpenACC
More Science, Less Programming

Learning Objectives

You will learn how to profile your application with NVIDIA® Nsight™ Systems and NVIDIA Tools Extension SDK (NVTX) API calls to find performance limiters and bottlenecks and apply incremental parallelization strategies. Throughout the labs, you will:

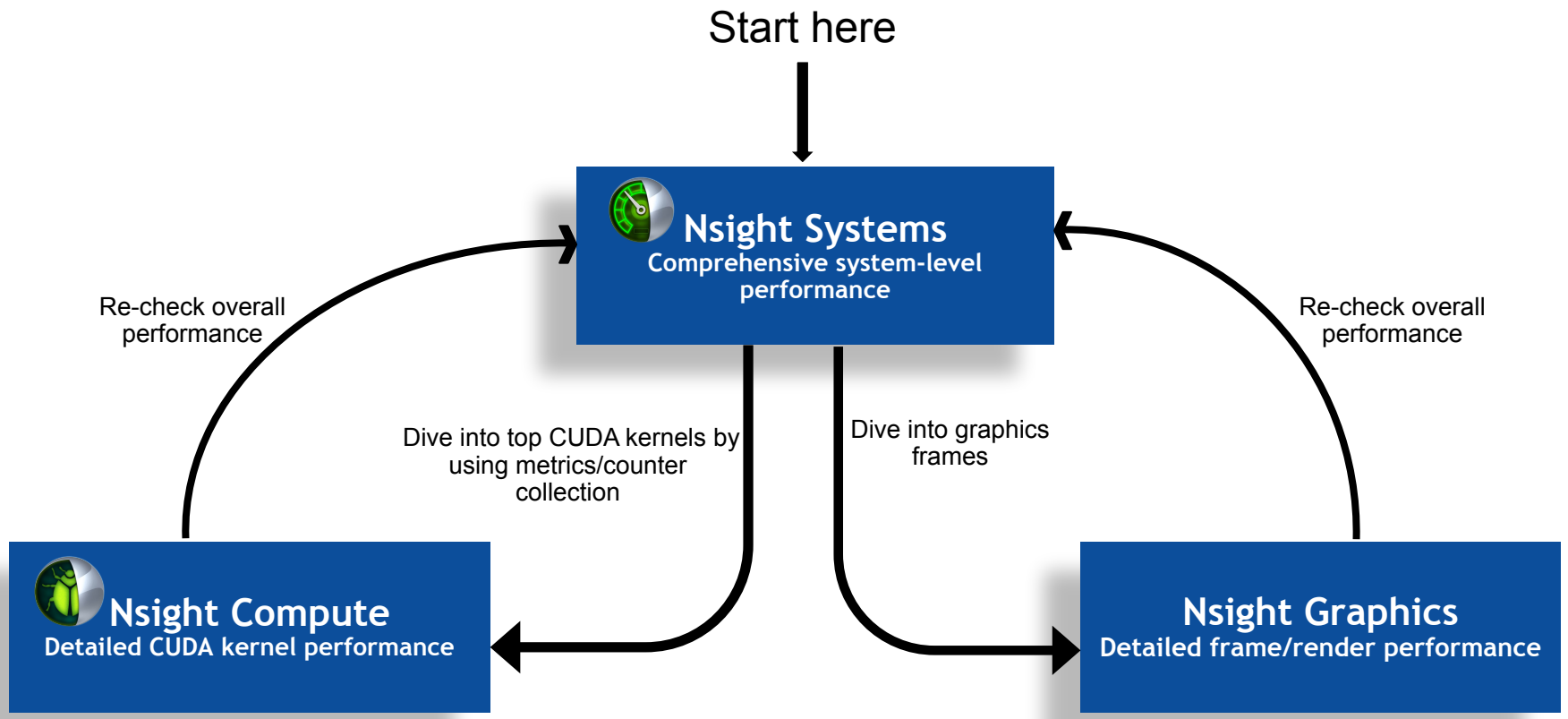
- Learn to follow a cyclical process (analyze, parallelize, optimize) to help you identify the portions of the code that would benefit from GPU acceleration and apply parallelization strategies and optimization techniques to see additional speedups and improve performance.
- Understand what a profiler is and which NVIDIA Nsight tool to choose in order to profile your application.
- Learn how to use **Nsight Systems** to identify issues in the application, such as an underutilized GPU device and unnecessary data movements, and to apply optimization strategies step-by-step to expose more parallelism and utilize computer's CPU and GPU.
- Learn how to use **Nsight Compute** to dive deep into the kernel and take the optimization to next level.

Introduction

What is profiling?

- Profiling is the first step in optimizing and tuning your application.
- Profiling an application helps you understand where most of the execution time is spent.
- With profiling, you gain an understanding of the application's performance characteristics and identify parts of the code that present opportunities for improvement.
- Profiling enables you to find hotspots and bottlenecks in your application so you can decide where to focus your optimization efforts.

Nsight Tools Workflow



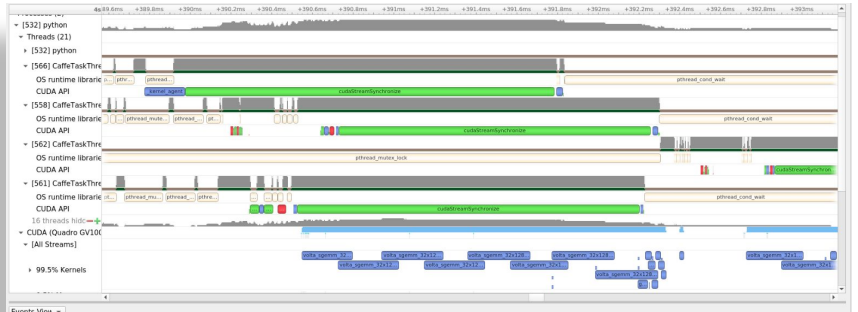
Nsight Systems

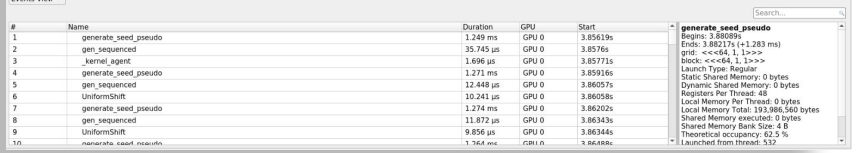
System profiler

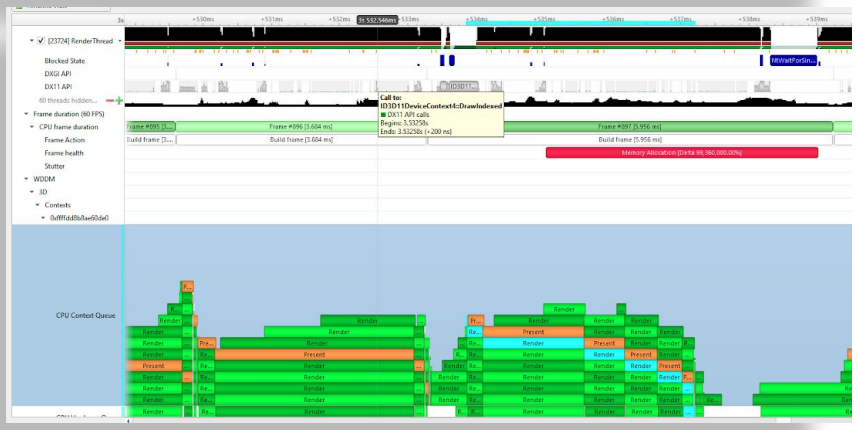
Key Features:

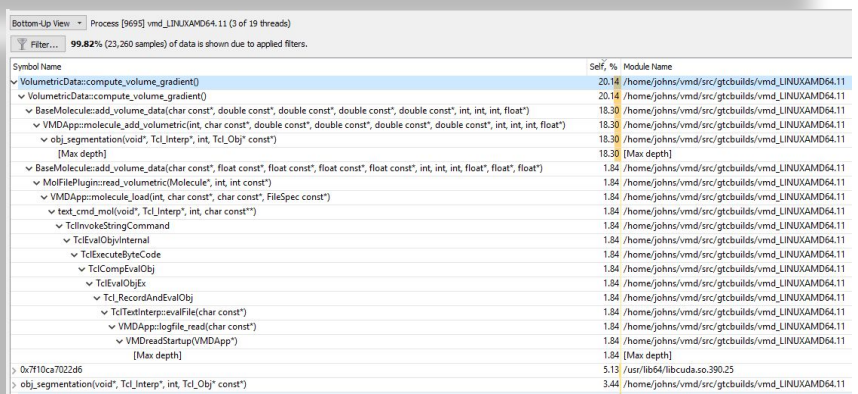
- System-wide application algorithm tuning
  - Multi-process tree support
- Locate optimization opportunities
  - Visualize millions of events on a very fast GUI timeline
  - Visualize gaps of unused CPU and GPU time
- Balance your workload across multiple CPUs and GPUs
  - CPU algorithms, utilization, and thread state
    GPU streams, kernels, memory transfers, etc
- Command Line, Standalone, IDE Integration

Docs/product: https://developer.nvidia.com/nsight-systems









OpenACC
More Science, Less Programming

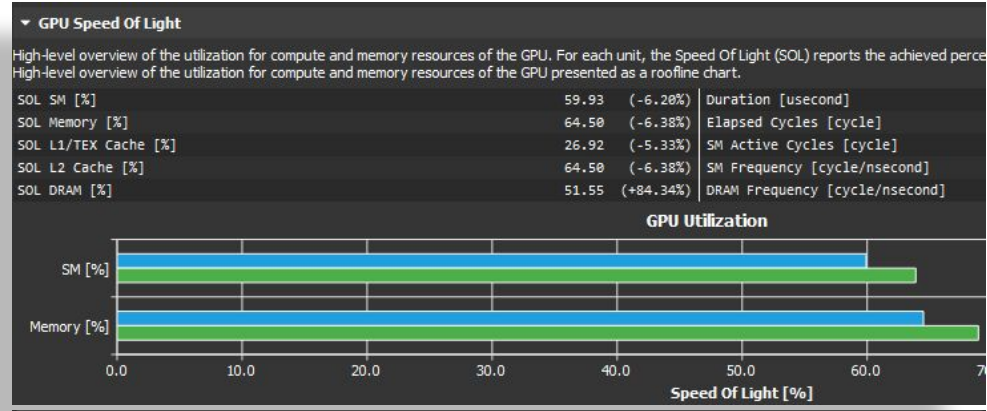
Nsight Compute

Kernel Profiling Tool

Key Features:

- Interactive NVIDIA CUDA® API debugging and kernel profiling
- Fast data collection
- Compare performance metrics across different runs
- Fully customizable (programmable UI/guided analysis)
- Command Line, Standalone, IDE Integration

Docs/product: https://developer.nvidia.com/nsight-compute



inst_executed [inst]	63,021,056 (284 instances)
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum	0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum	0
l1tex__data_bank_reads.avg.pct_of_peak_sustained_elapsed [%]	9.66
l1tex__data_bank_writes.avg.pct_of_peak_sustained_elapsed [%]	3.23
l1tex__data_pipe_lsu_wavefronts.avg.pct_of_peak_sustained_elapsed [%]	46.16
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_read.sum	25,165,824
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_read.sum.pct_of_peak_sustained_active [%]	40.75
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_write.sum	2,097,152
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_write.sum.pct_of_peak_sustained_active [%]	3.40
l1tex__data_pipe_tex_wavefronts.avg.pct_of_peak_sustained_elapsed [%]	0
l1tex__f_wavefronts.avg.pct_of_peak_sustained_elapsed [%]	0.00
l1tex__lsu_writeback_active.avg.pct_of_peak_sustained_elapsed [%]	42.59
l1tex__lsu_writeback_active.sum [cycle]	27,803,648
l1tex__lsu_writeback_active.sum.pct_of_peak_sustained_active [%]	45.03
l1tex__lsuin_requests.avg.pct_of_peak_sustained_elapsed [%]	66.00
l1tex__m_l1tex2xbar_req_cycles_active.avg.pct_of_peak_sustained_elapsed [%]	3.40
l1tex__m_l1tex2xbar_write_bytes.sum [Mbyte]	4.19
l1tex__m_l1tex2xbar_write_bytes_mem_global_op_red.sum [byte]	0

@P0 EXIT	6	108	49	1,404,672
IADD3 R7, P2, R0, UR7, RZ	7	177	95	1,401,344
IADD3 R6, P1, R4, UR4, RZ	8	30	9	1,401,344
ISETP.GE.U32.AND P0, PT, R7, UR5, PT	8	11	5	1,401,344
IADD3.X R8, R2, UR8, RZ, P2, !PT	8	14	8	1,401,344
IMAD.X R7, RZ, RZ, R5, P1	9	9	0	1,401,344
ISETP.GE.U32.AND.EX P0, PT, R8, UR6, PT, P0	9	106	35	1,401,344
STG.E.U8 [R6.64], R3	8	116	75	1,401,344
@P0 EXIT	8	92	33	1,401,344
IADD3 R8, P2, R0, UR9, RZ	9	45	14	1,397,120
IADD3 R6, P1, R6, UR4, RZ	9	248	145	1,397,120
ISETP.GE.U32.AND P0, PT, R8, UR5, PT	9	57	17	1,397,120
IADD3.X R8, R2, UR12, RZ, P2, !PT	9	11	4	1,397,120
IMAD.X R7, RZ, RZ, R7, P1	9	7	0	1,397,120
ISETP.GE.U32.AND.EX P0, PT, R8, UR6, PT, P0	9	94	15	1,397,120
STG.E.U8 [R6.64], R3	8	104	61	1,397,120

OpenACC
More Science, Less Programming

Hands-On

The lab has 3 sections:

Section 1

- Overview of Nsight profiler tools
- Optimization steps to parallel programming

Section 2

- Comprises 5 labs presenting porting and the optimization cycle of a serial application to the GPU using the OpenACC programming model and using NVIDIA profiling tools to find bottlenecks and hotspots

Section 3

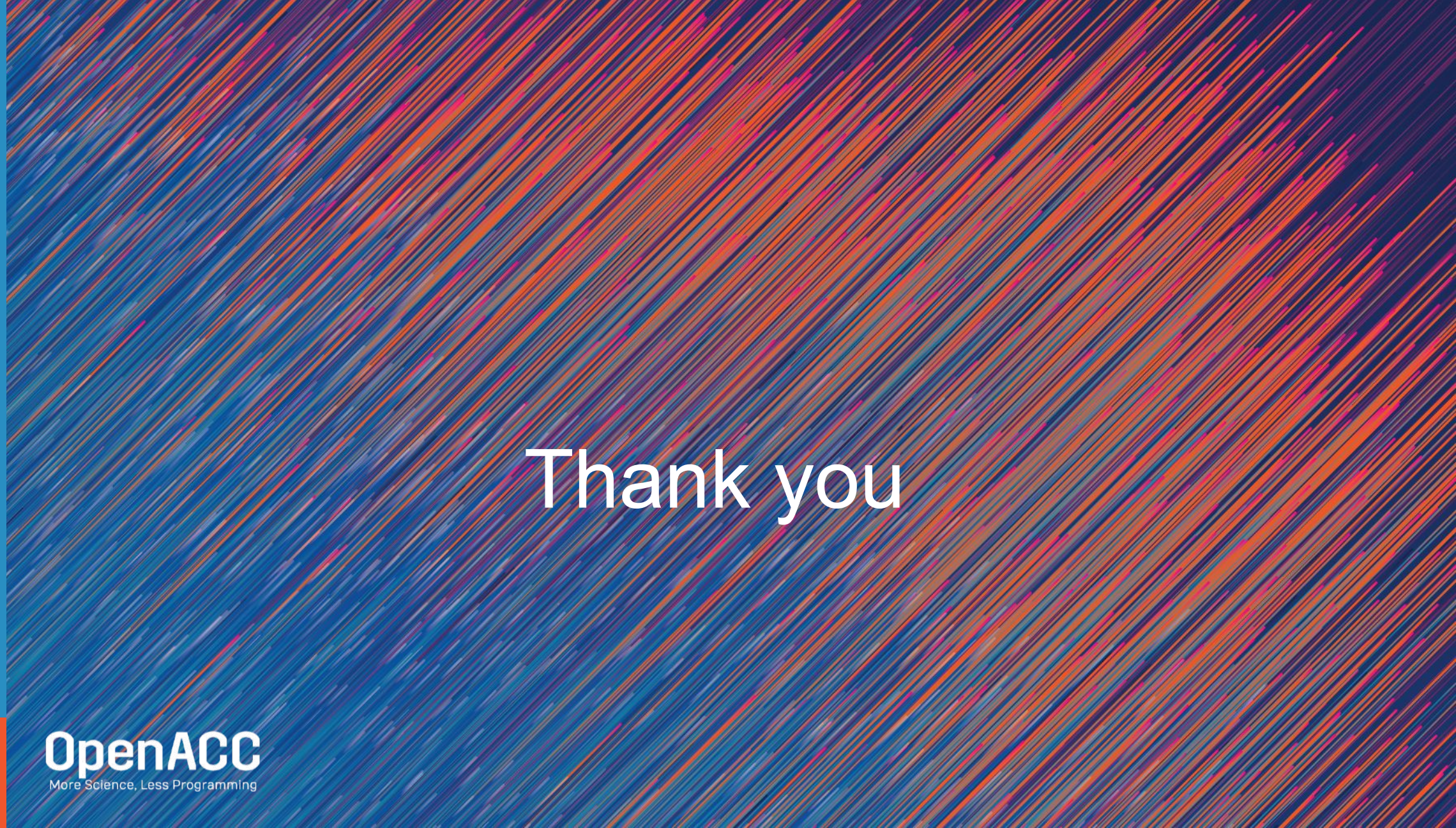
- Comprises one lab presenting porting and the optimization cycle of a serial application to the GPU using CUDA and using NVIDIA profiling tools to find bottlenecks and hotspots
- Advanced notebook covering GPU metrics and tips on profiling an MPI code with Nsight Systems

Resources and Links

- Additional resources
  - NVIDIA Nsight Systems
  - NVIDIA Nsight Compute
  - Open Hackathons technical resource page
  - Open Hackathons GitHub Repository

- Join the OpenACC and Hackathons Slack channel

- Licensing


Thank you
OpenACC
More Science, Less Programming